Banking AI System - Comprehensive Test Cases

1. 🏦 Transaction History Tests

Test the transaction_history *functionality*

Basic Transaction History

- show me my last 10 transactions
- tell me my recent transactions
- transaction history
- show me my activities
- list my last 5 transactions

Month-Specific Transactions

- show me my transactions in May
- transactions for June
- what did I do in April?
- July transactions
- show me January activity

Expected: Real transaction data with dates, descriptions, amounts, categories

---

2. 💰 Balance Inquiry Tests

Test the balance_check *functionality*

Basic Balance Checks

- what's my balance?
- how much money do I have?
- show me my account balance
- current balance
- how much do I have in my account?

Goal-Related Balance Checks

- do I have enough for a 1000000 PKR purchase?

- can I afford 50000 PKR?
- how much more do I need to reach 1000000?
- can I buy something that costs 25000?

Expected: Current balance with currency, affordability analysis for goals

---

3. 📊 Spending Analysis Tests

Test the spending_patterns *and* spending_breakdown *functionality*

General Spending Patterns

- how much do I spend on average?
- analyze my spending patterns
- what are my spending habits?
- show me my spending trends

Time-Based Spending Analysis

- what caused my high spending in May?
- why did I spend so much last month?
- break down my May expenses
- what drove my spending in April?

Expected: Spending breakdowns by category, time periods, patterns analysis

---

4. 🔄 Monthly Comparison Tests

Test the monthly_comparison *functionality*

Spending Comparisons

- did I spend more in May than April?
- am I spending more this month than last month?
- compare my spending across months
- how does my current spending compare to May?
- am I spending more right now than I did in April?

Expected: Month-by-month comparison with specific amounts and analysis

---

5. 🏷️ Category Spending Tests

Test the `category_analysis` *functionality*

Specific Category Analysis

- `how much did I spend on food this month?`
- `my entertainment expenses`
- `what did I spend on transport?`
- `dining expenses in May`
- `how much on Uber rides?`

Expense Cutting Queries

- `what expenses can I cut?`
- `which categories should I reduce spending on?`
- `help me cut my entertainment costs`
- `where can I save money?`

Expected: Category breakdowns with amounts, percentages, cutting suggestions

---

6. 🎯 Savings Planning Tests

Test the `savings_planning` *functionality*

Goal Planning

- `I want to save 1000000 PKR by August`
- `help me save for a car that costs 500000`
- `what can I do to reach 2000000 by December?`
- `how can I save 100000 in 6 months?`

Expected: Savings strategies, timeline analysis, expense cutting recommendations

---

7. 💸 Money Transfer Tests

Test the `transfer_money` *functionality*

Complete Transfer Requests

- `transfer 1000 PKR to John`

- send 500 to Ahmed
- I want to transfer 2000 PKR to Sarah
- pay 1500 to Ali

Incomplete Transfer Requests (should ask for missing info)

- transfer 1000 (missing recipient)
- send money to John (missing amount)
- I want to transfer money (missing both)

Expected: Complete transfers process, incomplete ones ask for missing information

---

8. 🔗 Contextual Query Tests

Test follow-up and contextual understanding

Follow-up Questions (after showing transactions)

1. First: show me my May transactions
2. Then: which ones were from restaurants?
3. Then: show me the highest amounts
4. Then: was that Netflix payment on the 15th?

Expected: AI should understand context and filter previous results

---

9. 🤖 General Assistant Tests

Test the direct_answer functionality

Greetings and General Questions

- hello
- hi there
- what can you do?
- help me
- good morning
- what services do you offer?

Expected: Friendly responses explaining capabilities, no database queries

---

10. 🚨 Error Handling Tests

Test system robustness

Invalid Requests

- `transfer -500 to John` (negative amount)
- `show me transactions for February 2030` (future date)
- `how much did I spend on unicorns?` (invalid category)

Expected: Graceful error handling with helpful messages

---

11. 🔄 Mixed Query Tests

Test complex, multi-part requests

Complex Queries

- `I spent a lot in May, what caused it and how can I reduce spending next month?`
- `Compare my food spending in April vs May and suggest ways to save`
- `Show me my entertainment expenses and tell me if I can afford a 50000 PKR purchase`

Expected: Comprehensive analysis addressing all parts of the question

---

12. 📱 Edge Cases

Test boundary conditions

Boundary Testing

- `show me my last 1 transaction`
- `transactions for a month with no activity`
- `transfer 0.01 PKR to someone`
- `show me 100 transactions` (at the limit)

Expected: System handles edge cases gracefully

🧪 Testing Instructions

For Each Test:

1. **Send the query** to the banking bot

2. **Check the response** matches expected functionality

3. **Verify data accuracy** (real transactions, correct calculations)

4. **Note response time** (should be reasonable)

5. **Test conversation flow** (memory and context work)

Success Criteria:

✅ **Correct routing** - Right function called for each query type
✅ **Real data** - Actual transactions/balances, not generic responses
✅ **Natural language** - Conversational, not robotic responses
✅ **Context awareness** - Remembers conversation flow
✅ **Error handling** - Graceful failures with helpful messages
✅ **Performance** - Responds within reasonable time

Red Flags:

❌ Generic "technical difficulties" messages
❌ Wrong function routing (balance check for transaction request)
❌ JSON serialization errors
❌ Crashes or timeouts
❌ Memory leaks or session issues

📊 Test Results Template

Test Session Results
Date: [DATE]
Account: [ACCOUNT]

Transaction History: ✅/❌
- Basic requests: ✅/❌
- Month filtering: ✅/❌
- Limit handling: ✅/❌

Balance Inquiry: ✅/❌
- Simple balance: ✅/❌
- Goal planning: ✅/❌

Spending Analysis: ✅/❌
- Patterns: ✅/❌
- Breakdowns: ✅/❌
- Comparisons: ✅/❌

Money Transfers: ✅/❌
- Complete transfers: ✅/❌
- Missing info handling: ✅/❌

Overall System: ✅/❌
- Response quality: ✅/❌
- Performance: ✅/❌
- Error handling: ✅/❌

Start with the **Transaction History** and **Balance Inquiry** tests since those are the most fundamental, then work your way through the more complex features!